```
% Week 4 code 151A
% Find root of function f(x)=cos(x)-x^3 using Newton's method and Secant method
clc;
clear all;
%% Parameters
tol = 1e-5; % error tolerance
N0 = 500; % maximum number of iterations
p0 = 1; % starting point
f = @(x) exp(x)+2^(-x)+2*cos(x)-6;
p_root=1.82938360193385;
%% Newton's method
f_diff = @(x) exp(x)-log(2)*2^(-x)-2*sin(x);
[iter, p] = Newton(f, f_diff, tol, N0, p0,p_root);
fprintf('Newton''s method:\n')
fprintf('Iteration number = %d \n', iter);
fprintf('p = %.11f \n',p);
fprintf('f(p) = %.11f \n\n', f(p));
%% Secant method
p1 = 1.5;
[iter, p] = Secant(f, tol, N0, p0, p1,p_root);
fprintf('Secant method:\n')
fprintf('Iteration number = %d \n', iter);
fprintf('p = %.11f \n',p);
fprintf('f(p) = %.11f \n\n', f(p));
%% Algorithms
function [iter, p] = Newton(f, f_diff, tol, N0, p0,p_root)
j = 1;
p = p0;
while j < N0
    y = f(p);
    y_diff = f_diff(p);
    % always a good idea to add checks
    if abs(y_diff) < 1e-12
        error('dividing by zero')
    end
    p_next = p - y / y_diff;
    if abs(p-p_root)<tol;
        break;
    end
    j = j + 1;
    p = p_next;
end
p = p_next;
iter = j;
end
function [iter, p] = Secant(f, tol, N0, p0, p1,p_root)
j = 1;
p = p1;
p_prev = p0;
while j < N0
    y = f(p);
    y_prev = f(p_prev);
    % always a good idea to add checks
    if abs(y - y_prev) < 1e-12
        error('dividing by zero')
    end
    p_next = p - y * (p - p_prev) / (y - y_prev);
    if abs(p-p_root)<tol;
```

```
        break;
    end
    j = j + 1;
    p_prev = p;
    p = p_next;
end
p = p_next;
iter = j;
end
```